UROP1100E

Fall 2021, HKUST

Analysis of Hierarchical Clustering Methods

Supervisor's Name: GOLIN Mordecai Jay

Student's Name: RUBAB Tamzid Morshed

Student ID: 20672457

Abstract

In this report, I analyze two hierarchical clustering algorithms for graphs drawn from planted partition model. In this model, nodes are divided into k clusters at first, and intra-cluster edges are added independently with some probability p and inter-cluster edges are added independently with probability q, where $p > q$. For the first algorithm, we assume that the partitions (clusters) are known. In this case, we can assume a generalized version of planted partition (intra-cluster probabilities can be anything larger than q). And for the second one, the partitions are unknown. The former achieves $\frac{3k-2}{3k}$ approximation and the latter asymptotically finds at least 0.5-approximation.

1 Introduction

Given a weighted graph $G = (V, E, W)$ and a tree T whose set of leaves is V, we define

$$\text{cost}_G(T) = \sum_{\{i,j\} \in E} w_{i,j} |\text{leaves}(T[i \vee j])|$$

And

$$\text{rev}_G(T) = \sum_{\{i,j\}\in E} w_{i,j}\,(n - |\text{leaves}(T[i\vee j])|)$$

Given a weighted graph G, our goal is to find tree T that minimizes the cost function or equivalently maximizes the revenue function. However, as stated in the paper [1], it is an np-hard problem. Later, various approximation algorithm was found for the revenue maximizing problem. The paper [3] gives a 0.585-approximation algorithm i.e. given graph G, this algorithm can find T such that $\mathbb{E}(\text{rev}_G(T)) \geq 0.585 \max_T \text{rev}_G(T)$.

In this report, we will work with random graphs (drawn from some known distribution). In this case, we are mostly interested in the expected revenue, which is defined as

$$\mathbb{E}(\text{rev}_G(T)) = \sum_{\{i,j\}\in E} w_{i,j}\mathbb{P}(\text{edge between } i \text{ and } j)\,(n - |\text{leaves}(T[i\vee j])|)$$

2 Assumptions

Given $k > 1$, N divisible by k, and $0 < q < 1$, we define a family of distributions called (n, q, k) planted partition model that contains all possible distributions described as follows:

We create a set of n nodes V and divide them into k groups, each containing $\frac{n}{k}$ nodes. For any pair of nodes (u, v), if they are in the same group, we add an edge between them with some probability $> q$ (this probability is fixed initially); otherwise we add an edge between them with probability q. Each edge will have a weight of 1.

Note that this model is family of distributions because there are infinitely many possible ways of assigning the intra-cluster probabilities. To get one distribution from this model, we need to initially set those probabilities (different edge can have different probabilities and each one is independent).

3 Random Algorithm for Known Partitions

In this section we assume that we know which node belong to which group and the probability of each edge. Then we use the following algorithm:

In the root node, we split the graph into those k groups. Subsequently, we use random splitting. In particular, the root node has k children and subtree rooted at each of these children is binary and generated using the random algorithm.

From the paper [1], we have the following result:

Given a graph $G = (V, E)$ and its partitions $T_1, T_2, \cdots, T_k$ (drawn from a fixed distribution D in the (n, q, k) planted partition model), we define a graph H as follows: Keep the same nodes and edges as G and add a weight of $w(i, j) = \mathbb{P}(\text{edge between } i \text{ and } j) - q$ to edge (i, j) for each $i, j \in V$. Now, fix a binary tree T with leaves V. Then $\mathbb{E}_D(\text{cost}_G(T)) = \frac{1}{3}q(n^3 - n) + \text{cost}_H(T)$.

This means that the problem of minimizing the expected cost of G is equivalent to the problem of minimizing the cost of H. But we do not know how to minimize the cost of H (in polynomial time), so we still look for approximation algorithms. Meanwhile, here we rewrite this equation in terms of revenues and prove the following result:

Lemma 1: Any algorithm that gives $0 < \alpha < 1$ approximation on maximum revenue problem for H, gives at least α approximation for maximum expected revenue problem for G.

Proof. First note that

$$\text{rev}_G(T) = n \sum_{\{i,j\} \in E} w_{i,j} - \text{cost}_G(T)$$

Thus combined with the previous result we have,

$$\begin{aligned}
& -\mathbb{E}(\text{rev}_G(T)) + n \sum_{\{i,j\} \in E} \mathbb{P}(\text{edge between } i \text{ and } j) = \frac{1}{3}q(n^3 - n) - \text{rev}_H(T) + n \sum_H w_{i,j} \\
\implies & -\mathbb{E}(\text{rev}_G(T)) + n \sum_{\{i,j\} \in E} \mathbb{P}(\text{edge between } i \text{ and } j) = \frac{1}{3}q(n^3 - n) - \text{rev}_H(T) \\
& \qquad + n \sum_H (\mathbb{P}(\text{edge between } i \text{ and } j) - q) \\
\implies & -\mathbb{E}(\text{rev}_G(T)) = \frac{1}{3}q(n^3 - n) - \text{rev}_H(T) - n\binom{n}{2}q \\
\implies & \mathbb{E}(\text{rev}_G(T)) = \text{rev}_H(T) + \frac{1}{6}n(n-1)(n-2)q
\end{aligned}$$

Therefore, maximizing revenue of H is also equivalent to maximizing expected revenue of G.

Let $T_{\max}$ be a tree maximizing the revenue and T_α be a tree such that

$$\text{rev}_H(T_\alpha) \geq \alpha \text{rev}_H(T_{\max})$$

Then since $\frac{1}{6}n(n-1)(n-2)q > 0$, we have

$$\mathbb{E}(\text{rev}_G(T_\alpha)) = \text{rev}_H(T_\alpha)+\frac{1}{6}n(n-1)(n-2)q \geq \alpha\text{rev}_H(T_{\max})+\frac{1}{6}n(n-1)(n-2)q \geq \alpha\mathbb{E}(\text{rev}_G(T_{\max}))$$

□

Note that the inter-cluster edges in H have 0 weight, so they can be removed from the graph. Thus H has k connected components. Consider a tree T' where the root node has k children (corresponding to the k partitions $T_1, \cdots, T_k$), and each subtree rooted at those k nodes is binary (shown in left of the figure below). Note that this tree has same revenue as the binary tree shown in right (because there is no edge between T_i and T_j for $i \neq j$).

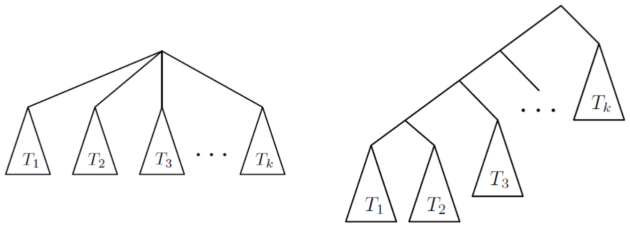

Figure 1: $T_1, \cdots, T_k$ are binary trees

So, instead of working with the binary tree, we will simply split the k connected components of H in the root node. Then we will generate the binary tree T_i (for each i) using the random algorithm.

Lemma 2: Let OPT be the optimal revenue for H. The process described above has expected revenue of at least $\frac{3k-2}{3k}OPT$.

Proof. From the paper [3], we know that there exists a bisection that has revenue at least $0.5OPT$.

Let W be the total weight of H. Note that in the maximum uncut bisection the total weight of the uncut edges can be at most W. Thus the maximum revenue a bisection can have is $\frac{1}{2}nW$. Thus $\frac{1}{2}nW \geq \frac{1}{2}OPT \implies nW \geq OPT$.

Now in our k-partition, the total weight of the uncut edges is also W (as there is no inter-cluster edge in H). Besides we know that the random algorithm generates at least $\frac{1}{3}\frac{n}{k}$ expected revenue for each edge in each of the k clusters. Thus the total expected revenue for the process is at least

$$(\frac{k-1}{k}n + \frac{1}{3}\frac{n}{k})W = \frac{3k-2}{3k}nW \geq \frac{3k-2}{3k}OPT$$

$\square$

Combining with lemma-1, we can say that the tree obtained from the process described before lemma-2 gives an expected approximation of at least $\frac{3k-2}{3k}$ for graph G generated from a distribution from the (n, q, k) planted partition model.

4 Average Linkage for Unknown Partitions

Now we consider the simple (n, p, q)-planted partition model, where each edge within a cluster exist with probability p (independently). We now do not know which node belong to which group. So we cannot simply split the k groups in the root node. Here, we analyze the performance of the average linkage for this kind of graphs.

Let G be a graph generated from this model. Let T_G be the tree generated from the average linkage algorithm. From [2] we know that

$$\text{rev}_G(T_G) \geq \frac{n-2}{3} \sum_{i,j} w_{ij}$$

Taking expectation over the distribution of G,

$$\begin{aligned}
\mathbb{E}(\text{rev}_G(T_G)) &\geq \frac{n-2}{3} \sum_{i,j} \mathbb{E}(w_{ij}) \\
&= \frac{n-2}{3} \left(\sum_{i,j \text{ in same cluster}} \mathbb{E}(w_{ij}) + \sum_{i,j \text{ not in same cluster}} \mathbb{E}(w_{ij}) \right) \\
&= \frac{n-2}{3} \left(\sum_{i,j \text{ in same cluster}} p + \sum_{i,j \text{ not in same cluster}} q \right) \\
&= \frac{n-2}{3} \left(\binom{\frac{n}{k}}{2} kp + \binom{k}{2} \left(\frac{n}{k}\right)^2 q \right) \\
&= (n-2) \left(\frac{1}{3} \left(\frac{n^2}{k^2} - \frac{n}{k} \right) \frac{k}{2} p + \frac{k-1}{2} \frac{n^2}{k^2} \frac{kq}{3} \right)
\end{aligned}$$

Now consider H generated from G as described in the previous section. We know that if T_G^* maximizes $\mathbb{E}(\text{rev}_G(T_G))$, then it also maximizes $\text{rev}_H(T_G)$. And note that H has k connected component and each component is a clique. So, by paper [1], any binary tree that first separates the k components give the best revenue. Let T_G^* be such a tree. Then

$$
\begin{aligned}
\mathbb{E}(\text{rev}_G(T_G^*)) &= \frac{1}{6}n(n-1)(n-2)q + rev_H(T_G^*) \\
&= \frac{1}{6}n(n-1)(n-2)q + \frac{1}{3}\left(\frac{n^3}{k^3} - \frac{n}{k}\right)(p-q)k + \binom{\frac{n}{k}}{2}\frac{k-1}{k}n(p-q)k \\
&= \frac{1}{6}n(n-1)(n-2)q + \left(\frac{1}{3}\left(\frac{n^3}{k^3} - \frac{n}{k}\right) + \binom{\frac{n}{k}}{2}\frac{k-1}{k}n\right)pk \\
&\qquad - \left(\frac{1}{3}\left(\frac{n^3}{k^3} - \frac{n}{k}\right) + \binom{\frac{n}{k}}{2}\frac{k-1}{k}n\right)qk \\
&= \frac{1}{6}n\left(n^2\frac{k-3}{k} + 4 + \frac{n}{k}(\frac{n}{k} - 3)\right)q + \left(\frac{1}{3}\left(\frac{n^3}{k^3} - \frac{n}{k}\right) + \binom{\frac{n}{k}}{2}\frac{k-1}{k}n\right)pk
\end{aligned}
$$

Now we compare $\mathbb{E}(\text{rev}_G(T_G^*))$ and $\mathbb{E}(\text{rev}_G(T_G))$. Note that

$$
\frac{1}{3}\left(\frac{n^2}{k^2} - \frac{n}{k}\right)\frac{k}{2}p + \frac{k-1}{2}\frac{n^2}{k^2}\frac{kq}{3} > \frac{1}{6}\left(n^2\frac{k-3}{k} + 4 + \frac{n}{k}(\frac{n}{k} - 3)\right)q
$$

Fix k such that $\frac{kq}{3} \geq p$. Then for large n,

$$
\mathbb{E}(\text{rev}_G(T_G)) \sim \left(\frac{1}{3}\left(\frac{n^3}{k^3} - \frac{n}{k}\right) + \binom{\frac{n}{k}}{2}\frac{k-1}{k}n\right)pk
$$

Thus given that $\frac{kq}{3} \geq p$, for large n, average linkage gives at least 0.5 approximation coefficient.

References

[1] Sanjoy Dasgupta. 2016. A cost function for similarity-based hierarchical clustering. In Proceedings of the forty-eighth annual ACM symposium on Theory of Computing (STOC '16). Association for Computing Machinery, New York, NY, USA, 118–127. DOI:`https://doi.org/10.1145/2897518.2897527`

[2] Benjamin Moseley and Joshua R. Wang. 2017. Approximation bounds for hierarchical clustering: average linkage, bisecting K-means, and local search. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). Curran Associates Inc., Red Hook, NY, USA, 3097–3106.

[3] Alon, N., Azar, Y. amp; Vainstein, D.. (2020). Hierarchical Clustering: A 0.585 Revenue Approximation. *Proceedings of Thirty Third Conference on Learning Theory*, in *Proceedings of Machine Learning Research* 125:153-162 Available from `https://proceedings.mlr.press/v125/alon20b.html`.